"FIRST-TRY" DNS CACHE POISONING WITH IPV4 AND IPV6 FRAGMENTATION

Or how to become the one in “one in 34 million”
WHERE WE’RE GOING

1. **Intro**
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
Travis (Travco) Palmer

- Security Research Engineer for Cisco Systems
- Offensive Security Certified Professional & Expert (OSCP & OSCE)
- Not a DNS/DNSSEC expert

Brian Somers

- Principal Engineer for Cisco Systems
- FreeBSD & OpenBSD developer alumnus
You did what?

- Found a more consistent way to poison the cache of DNS resolvers without man-in-the-middle
- Modified an IPv4 attack on DNS over UDP, reduced it from hundreds of iterations to plausibly one
- Extended the attack so that it bypasses all current recommendations
YES, WE DID DISCLOSE RESPONSIBLY

- Our team discovered this attack during a focused pentest engagement

- Our team disclosed to Cisco Umbrella

- Umbrella has been disclosing this to other DNS operators (ongoing) before DEF CON.
WHERE WE’RE GOING

1. Intro
2. **Background on DNS**
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
If the timing on a particular DNS request can be predicted, the reply only needs to be well structured and have a valid ID.

In 2008 Dan Kaminsky demonstrated 16 bits of entropy is not sufficient to prevent cache poisoning.

And can be performed off-path (source ports are predictable).
IDEAL POISONING SCENARIO

TARGET

PUPPET

CACHE

INTERNET

RESOLVER

ATTACKER

NAMESERVER

TRUST BOUNDARY
IDEAL POISONING SCENARIO

TARGET

PUPPET

RESOLVER

CACHE

INTERNET

NAMESERVER

ATTACKER

Resolver is v.x.y.z!
IDEAL POISONING SCENARIO

TARGET

PUPPET

INTERNET

CACHE

TRUST BOUNDARY

RESOLVER

ATTACKER

IDEAL POISONING SCENARIO

TARGET

PUPPET

INTERNET CACHE

RESOLVER

INTERNET

ATTACKER


TRUST BOUNDARY

Not Poisoned? Try, try again (just make sure it isn’t cached)
IDEAL POISONING SCENARIO

- TARGET
- PUPPET
- RESOLVER
- INTERNET
- NAMESERVER
- ATTACKER

IDEAL POISONING SCENARIO

TRUST BOUNDARY

TARGET

PUPPET

RESOLVER

CACHE

INTERNET

NAMESERVER

ATTACKER
IDEAL POISONING SCENARIO

TARGET

PUPPET

CACHE

RESOLVER

INTERNET

ATTACKER

TRUST BOUNDARY
IDEAL POISONING SCENARIO

TRUST BOUNDARY

TARGET

PUPPET

RESOLVER

CACHE

INTERNET

ATTACKER

NAMESERVER
DNS source ports aren’t predictable anymore.

To fake a DNS response off-path, a 16bit DNS identifier, and a UDP port number (16bit*) need to be guessed.
Enter DNS Security Extensions (DNSSEC)

Cryptographic key-based signing of DNS zones by parent zones, and signing of records by zones.

ICANN “.”
“.org” TLD
.defcon.org
www.defcon.org. IN A 162.222.171.206
Enter DNS Security Extensions (DNSSEC)

https://www.cloudflare.com/dns/dnssec/how-dnssec-works/
DNSSEC adds (most importantly):

- **Data origin authentication** - Verify that the data it received actually came from the zone it should have come from.

- **Data integrity** - Data cannot be modified in transit since records are signed by the zone owner with the zone's private key.
ICANN Calls for Full DNSSEC Deployment, Promotes Community Collaboration to Protect the Internet

LOS ANGELES – 22 February 2019 – The Internet Corporation for Assigned Names and Numbers

## QUICK AND DIRTY DNS PRIMER

<table>
<thead>
<tr>
<th>Region</th>
<th>DNSSEC Validates</th>
<th>Uses Google PDNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>14.95%</td>
<td>14.16%</td>
</tr>
<tr>
<td>Oceania</td>
<td>23.80%</td>
<td>5.61%</td>
</tr>
<tr>
<td>Americas</td>
<td>22.50%</td>
<td>12.88%</td>
</tr>
<tr>
<td>Europe</td>
<td>20.02%</td>
<td>9.33%</td>
</tr>
<tr>
<td>Africa</td>
<td>16.58%</td>
<td>28.61%</td>
</tr>
<tr>
<td>Asia</td>
<td>10.17%</td>
<td>13.11%</td>
</tr>
</tbody>
</table>

QUICK AND DIRTY DNS PRIMER

DNSSEC doesn’t:

- Sign Delegation NS and A resource records (RRs)
- Sign Glue Records

Origin of work: “Fragmentation Considered Poisonous”, Amir Herzberg and Haya Shulman, Published 2012
RR types, do not form RRsets. In particular, the TTL values among RR SIG RR with a common owner name do not follow the RRset rules described in [RFC218].

An RRSIG RR itself MUST NOT be signed, as signing an RRSIG RR would add no value and would create an infinite loop in the signing process.

The NS RRset that appears at the zone apex name MUST be signed, but

RR sets in the parent zone that delegate the name to the child zone’s name servers) MUST NOT be signed. Glue address RR sets associated with delegations MUST NOT be signed.

There MUST be an RRSIG for each RR set using at least one DNSKEY of each algorithm in the zone apex DNSKEY RRset. The apex DNSKEY RRset itself MUST be signed by each algorithm appearing in the DS RR set located at the delegating parent (if any).
QUICK AND DIRTY DNS PRIMER

Signing comes at a cost, especially with NSEC/NSEC3

$ dig +dnssec @dns-2.datamerica.com. gggg.defcon.org
Quick and dirty DNS Primer

Signing comes at a cost, even for NSEC3

$ dig +dnssec @dns-2.datamerica.com.

gggg.datamerica.com. 86400 IN SOA dns-2.datamerica.com. (username@username) Origin of work: "Fragmentation Considered Poisonous", Amir Herzberg and Haya Shulman, Published 2012

Quick and dirty DNS primer

Origin of Text:
WHERE WE’RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
If a response becomes too big, it needs to be fragmented at the IP layer.

The DNS identifier and UDP port number are early in the IP payload.

For the second fragment, the only entropy is the IP identifier (IPID) in the header.

Origin of work:
“Fragmentation Considered Poisonous”, Amir Herzberg and Haya Shulman, Published 2012
The IP identifier (IPID) for IPv4 is 16 bits.

A significant portion of nameservers were found to have a single global counter for IPID.
IDEAL POISONING SCENARIO

TARGET

PUPPET

ATTACKER

INTERNET

CACHE

RESOLVER

DNS Cache

IP Fragment Cache

TRUST BOUNDARY
IDEAL POISONING SCENARIO
IDEAL POISONING SCENARIO

TRUST BOUNDARY

TARGET

PUPPET

RESOLVER

INTERNET

CACHE

0

ATTACKER

NAMESERVER
IDEAL POISONING SCENARIO

TARGET

INTERNET

IPID = 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, etc...

PUPPET

ATTACKER

RESOLVER

CACHE

TRUST BOUNDARY

We might need to repeat this (Patience/Avoiding Cache)
IDEAL POISONING SCENARIO

TRUST BOUNDARY

TARGET

PUPPET

RESOLVER

INTERNET

ATTACKER

CACHE

IPID = 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, etc...
IDEAL POISONING SCENARIO

TARGET

TRUST BOUNDARY

CACHE

IPID = 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, etc...

INTERNET

RESOLVER

PUPPET

ATTACKER

NAME SERVER
IDEAL POISONING SCENARIO

TRUST BOUNDARY

TARGET

PUPPET

CACHE

RESOLVER

INTERNET

NAMESERVER

ATTACKER
IDEAL POISONING SCENARIO
IDEAL POISONING SCENARIO
DNSSEC Adds lots of signature records, but the authority (NS) and additional sections are always last.

Subdomain Injection, NS Hijacking, NS Blocking

Origin of work: “Fragmentation Considered Poisonous”, Amir Herzberg and Haya Shulman, Published 2012.
DNSSEC adds lots of signature records, but the authority (NS) and additional records are last.

Subdomain Injection, NS Hijacking, NS Blocking

<table>
<thead>
<tr>
<th>Requirements</th>
<th>DNS Poisoning (Section 4)</th>
<th>Name Server Blocking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domain Hijacking</td>
<td>Subdomain Injection</td>
</tr>
<tr>
<td>IP-ID</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>‘Fragmentable zone’</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>‘Poisonable zone’</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>‘Permissive or Island’</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NSEC3 opt-out</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RFC 4697</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MALICIOUSLY FORCING FRAGMENTATION

TARGET

PUPPET

RESOLVER

CACHE

INTERNET

ATTACKER

NAMESERVER

TRUST BOUNDARY

Spoofed ICMP fragmentation-required

IPID = 5002, 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, 5012
A decreasing number of deployed nameservers/OSs should be using sequential and global counters.

We can’t re-query things that get cached.

With IPv6, the IPID in the fragmentation extension header is 32bits, with a cache of 64 fragments:

- Realistic average ~34 million iterations
- Unrealistic ideal average ~17 million iterations

Origin of work: “Fragmentation Considered Poisonous”, Amir Herzberg and Haya Shulman, Published 2012
There has been some notice

- Prior to our engagement with Umbrella (April 2019), their implementation used IPv6 whenever possible, detected IPv4 fragments, and re-queried over TCP
- Workshop presentation at OARC 30 (Mid-May 2019)…
PERTINENT LIMITATIONS

... but the presentation wasn’t us...

- On IPv4, probability of spoofing $P_{s\_frag} = P_s \times 64000$
  - Probability is 64000 times larger than traditional cache poisoning
- On IPv6, $P_{s\_frag}$ is not changed
  - IPv6 Fragmentation ID is 32 bit, DNS ID is 16bit, port number is 16bit
- Fragmentation attack is effective only for IPv4
  - If IPv6 Fragmentation ID is random.
WHERE WE’RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. **IPID Inference**
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
There is a storied history of using IPID for Idle Scanning.

"Counting Packets Sent Between Arbitrary Internet Hosts", Jeffrey Knockel and Jedidiah R. Crandall, 2014
Two relevant changes to Linux Kernel:

- A patch that adds perturbation (2014)
- A patch that replaces per-destination IPID counters with “binned” counters (2014)

Origin of work:
“ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018
A patch that adds perturbation

author  🌟 Eric Dumazet <edumazet@google.com> 2014-07-26 08:58:10 +0200
committer 🌟 David S. Miller <davem@davemloft.net> 2014-07-28 18:46:34 -0700
commit  04ca6973f7c1a0d8537f2d9906a0cf8e69886d75 (patch)
tree   7f66f046e591ca2f0e58e67cbe19744d674796b4
tparent 545469f7a5d7f7b2a17b74da0a1bd0c1aea2f545 (diff)
download  linux-04ca6973f7.tar.gz

ip: make IP identifiers less predictable

In "Counting Packets Sent Between Arbitrary Internet Hosts", Jeffrey and Jedidiah describe ways exploiting linux IP identifier generation to infer whether two machines are exchanging packets.

When sending a packet, increments IPID by a normal distribution between 1 and the kernel ticks elapsed

Origin of work:
“ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018
A patch that replaces per-destination IPID counters with “binned” counters

Origin of work:
“ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018

A patch that replaces per-destination IPID counters with “binned” counters

```c
atomic_t *ip_idents__read_mostly;
-EXPORT_SYMBOL(ip_idents);
#define IP_IDENTS_SZ 2048u
+struct ip_ident_bucket {
    atomic_t id;
+    u32 stamp32;
+};
+static struct ip_ident_bucket *ip_idents__read_mostly;
```

One of 2048 “bins” (IP_IDENTS_SZ default)


Origin of work:
“ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018
MADE ONIS (2018) MISSED ONUS

ONIS: ONIS is Not an Idle Scan

- Use the IP-space of IPv6 for source addresses
- Find hash collisions between destination addresses by seeing the increment from zombie to target
- Get "under" perturbations (for most systems this timing is ~10ms but may be as low as ~0.66ms)

Origin of work: “ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018
Once a collision is found, start using the “zombie” for Not an Idle Scan

But wait... wasn't the only thing preventing DNS Fragment Poisoning the difficulty of guessing the IPID?
WHERE WE’RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
Much like with ONIS, start by finding collisions.

But wait... didn’t you say something about IPv6 being used?

This additive work is ours, diagram edited from:
"ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path", Zhang, Knockel, and Crandall. Published 2018
- Works for IPv6 when fragmented
- What about getting address space?

This additive work is ours, diagram edited from: “ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path”, Zhang, Knockel, and Crandall. Published 2018
What about getting address space?

All AWS Virtual Private Clouds (including free tier)
What about getting address space?

All AWS Virtual Private Clouds (including free tier) can add a /64 IPv6 CIDR (18,446,744,073,709,551,616 hosts)
Wait, why is this better than finding global IPID nameservers?

- Broader selection, all ‘recent’ Linux kernels (>3.16 – Aug 3 2014)
- Binning acts in our favor, ~99.95% of other hosts will not change the IPID (2047/2048)
Being a downstream puppet is trivial for public resolvers and organizations and individuals are increasingly relying on them.

... and organizations and individuals are increasingly relying on them.
WarGames and Waiting Games

Nameserver uptime is in the attacker's favor

- The secret key for hashing destination addresses only changes at reboot – so... \( \infty \) uptime on nameservers?
- Wait for times of least monitoring
- Accumulate collisions (matches) for multiple nameservers and resolvers
- Perform multiple short-duration cache attacks (for cases where we can specify timeout)
Unrealistic blind attacks are now plausible "in one shot"

- If this is doable in a single hit:
  - Maybe don’t need a downstream puppet
    - Anticipate automated clients (e.g. cron-jobs) does your blueteam work midnights?
  - Maybe the puppet can be passive
    - Attempt to poison common requests when they go out of cache
  - Maybe poison isn’t the purpose
    - Use NS Blocking to kill communication to all nameservers

NS Blocking work: “Fragmentation Considered Poisonous”, Amir Herzberg and Haya Shulman, Published 2012
1. Pick targets (Resolver, Domain & Linux Nameserver)
2. Determine Resolver’s public IP address for requests
3. Evaluate Domain responses, see what can be poisoned
4. Find a bucket collision between the attacker and Resolver addresses using the ONIS technique
5. ---- Wait until you feel like it ----
6. (Optional) trick Nameserver into lowering the PMTU and force DNS fragmentation
7. Query the Nameserver to get the IPIID just before a known request is sent to the Resolver (probably with a puppet)
8. Send a spoofed 64 fragment sequence based a known IPID as described in Fragmentation considered Poisonous
WHERE WE’RE GOING

1. Intro
2. Background on DNS
3. Fragmentation Attacks
4. IPID Inference
5. The Attack (agnostic to IPv4 and IPv6)
6. Mitigations
Identify and handle fragments as 'Suspect' (non-trivial)

- What Umbrella was going to deploy for IPv4
- Handle fragments in pre-assembly
  - Content in later UDP fragment should be untrusted
- Trigger re-queries at a higher layer over TCP
- Issue: IPv6 headers
  - IPv6 extension headers might not exist, may be in any order
FOR RESOLVERS

Implement “Flag Day 2020+” plans now

- Date TBD
- Cap EDNS (Extended DNS) bufsize solicitation at ~1220
  - More feasible with elliptic-curve RRSIGs
  - Avoid IPv6 fragmentation
- Drop all fragments (including IPv6)
- Re-query larger payloads over TCP

https://dnsflagday.net/2020/
FOR RESOLVERS

Be alerted (or very afraid) of unsolicited ECHO responses

- Indications of this attack are... limited. But one can still make an alert for what little warning there is
  - A large volume of unsolicited, fragmented, IPv6 ICMP Echo replies during collision-finding may be the only indication

- Though, this attack could be performed with sufficient IPv4 address space, or other protocols that allow for sufficiently tight-timing of responses.
Have you tried turning it off and on again?

- For a host running modern Linux, changing the key used to hash destination addresses would silently remove any known collisions.

- Obviously, even if this is done without a reboot - not ideal (traffic volume)
Limit EDNS over UDP ("Flag Day 2020+")

- Not really "Compliant" yet, but can still serve large responses over TCP
- Speed is important, but may be best left to the resolvers, most things can be cached

https://dnsflagday.net/2020/
Disable, fuzz, or limit what ICMPs you respond to

- There are good reasons for responding to ICMP ECHOs (especially as a backbone-of-DNS) but... maybe not fragmented pings
- Handle ICMP separately with a non-kernel process (IPID)
- Limit speed of replying to ICMPs
- ... but then again, ICMP *isn’t* the only way this attack could be done
In order of increasing difficulty…

Roll your own Kernel (sorry in advance) or use another

- Change IP_IDENTS_SZ to something much higher than 2048, recompile.

- Alternatively, use a kernel that is properly per-destination and take the performance hit

https://xkcd.com/303/
Deploy DNSSEC ... and do it with good signing keys

- Although DNSSEC produces longer replies (fragmentation), it also prevents outright tampering with A records.
- If you have a weak key your replies would be fragmentable, and the signing could be broken.

Key Signing Key (KSK) Which authenticates

Zone Signing Key (ZSK) Makes RRSIGs for records

A www.defcon.org w.x.y.z
RRSIG A 10 3 2419200 2019 ...
Deploy DNSSEC … and do it with good signing keys

- Although DNSSEC produces longer replies (fragmentation), it also prevents outright tampering with A records.
- If you have a weak key your replies would be fragmentable, and the signing could be broken.

6.1. Key lengths and algorithms

Key pairs are required to be of sufficient length to prevent others from determining the key pair’s private key using crypto-analysis during the period of expected utilization of such key pairs.

The current RZ ZSK key pair(s) is an RSA key pair, with a modulus size of at least 1024 bits.
FOR EVERYBODY ELSE...

DON’T PANIC
(well... maybe panic a little)
ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path
Zhang, Knockel, and Crandell

Fragmentation Considered Poisonous - Herzberg and Shulman

ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path
- Zhang, Knockel, and Crandell

IP_IDENTS_SZ in current Linux kernel:

The O.G. Kaminsky:

Two Main Papers:
Fragmentation Considered Poisonous - Herzberg and Shulman
ONIS: Inferring TCP/IP-based Trust Relationships Completely Off-Path
- Zhang, Knockel, and Crandell

IP_IDENTS_SZ in current Linux kernel:

Other resources in order of appearance:
https://www.cloudflare.com/dns/dnssec/how-dnssec-works/
https://indico.dnsoarc.net/event/31/contributions/692/attachments/660/1115/fujiwara-5.pdf
“Counting Packets Sent Between Arbitrary Internet Hosts”, Jeffrey Knockel and Jedidiah R. Crandall, 2014
https://dnstflagday.net/2020/